for the purpose of patentably distinguishing over the cited prior art. The applicant also submits new and more comprehensive substantive arguments in order to traverse the examiner's prior art rejection of the claims.

It is applicant's contention that none of the claims are taught or suggested by the cited prior art, as will be discussed in detail below. The design time controls disclosed by Massena are used, not at run-time, but at design-time, for the purpose of adding text-based components to web pages. These pages are then stored on the web server, where they can be accessed by a remotely-located browser. The Massena patent does not teach or suggest generating web pages that have usable editing features built into the page and intended for use by the end user in his browser program. In fact, the Massena patent teaches away from such a feature by ignoring or removing the design time controls of Massena when a browser requests such a web page. Further, Massena contemplates editing only by the author on the client side, and it is clear from the from the context of Massena that editing can only occur off-line, and not during run-time.

The Pinard patent, although generally relevant to web page generation, adds nothing to Massena which would teach or suggest these features.

Applicant and his counsel would also like to interview with the Examiner in the short term to discuss the prior art rejections and applicant's arguments traversing the rejections, and possibly arrange for a demonstration of applicant's system as necessary.

II. BRIEF DESCRIPTION OF THE CLAIMED INVENTION

A. Overview

Applicant uses the term "Interactive Server Side Component" ("ISSC") to describe his software development. ISSC's are software components used to build web applications, as described in more detail below. In addition, applicant has developed a browser based editor to maintain the components on dynamic web pages. The ISSC's encapsulate dynamic page functions, including processing of user responses, on the server, whereby the set of components contained on each page can change on a per request basis.

In contrast to well known client based editors, applicant's editor works as a web application on the server. **This enables direct editing of fully functional application pages**

during execution of the application. Applicant submits that this feature is not taught or suggested by any of the art cited by the Examiner in the parent application, .

Applicant has presented 14 sets of claims having differing language and scope, which are organized as follows:

Base claim	Dependent claims	Subject
1	2-5	Software development system
6	7-8	Software development system
9	10-15	Method for generating documents
16	17-21	Method for implementing client server applications
22	23-25	Server computer
26	27-33	System for modifying documents
34	35-50	Method for generating a page
51	51-58	System for editing components
59	60-73	Software development system
74	75-89	Software development system
90	90-96	Editor
97	98-113	System for displaying dynamically generated documents
114	115-124	System for displaying dynamically generated documents
125	126-127	Editing method

These 14 sets of independent claims can be logically organized into 4 groups, as follows:

(I) Independent claims 1, 59, 90 and 125

These claims describe running a web application in a browser program with the ability to concurrently edit the application in order to allow the developer to simultaneously test and edit the application. For example, claim 1 is directed to a software development system and makes clear that the editor is server based by reciting "*an editor capable of directly operating on the pages displayed by the browser thereby allowing the user to work on a functional application during development*".

Claim 59 is also directed to a software development program and recites "*an editor program for editing dynamic web documents.*" The editor program is clearly capable of also running the application during editing since it includes "*first instructions for requesting the document generator to process a dynamic web document leading to a generated document*".

Claim 90 is directed to an editor for use with a web browser, "*the editor allowing the user to edit a document displayed by the browser*" while "*scripts contained in said document remain functional*".

Claim 125 is directed to a method for editing an application including a step of "*running the application, thereby generating a generated document*" as well as a step for "*displaying a view of the generated document.*"

(II) Independent Claims 22, 26 and 51

These claims describe a component editor which works as a web application. For example, claim 22 is directed to a computer running a browser application and recites two elements, namely *an editor* and *a page generator*, wherein the editor is "*operable within the web browser for inserting, deleting and modifying components on document templates.*"

Claim 26 is directed to a server computer-having "*first software program instructions including instructions for transforming a first document . . . into a second document having features which permit editing of the first document.*"

Claim 51 is directed to a system for editing components on document templates which includes first, second and third software programs for accomplishing specified functionality. The third program provides instructions "*for modifying document templates in order to perform an editing function.*" The first and the second programs communicate by sending document requests and obtaining documents.

(III) Independent Claims 6, 9, 16, 34, 74 and 97

These claims stress the interactive and dynamic nature of the components. For example, claim 6 recites a server computer with "*a plurality of components . . . including components that react interactively on user input by executing instructions on the server.*" Similarly, claim 97 recites "*a plurality of components for execution on the server computer . . . including second program instructions*" wherein second program instructions are initiated by user interaction with a component.

Claim 9 is directed to a method for generating documents using components that react interactively, wherein the server executes steps for "*assigning a unique identifier*" and for "*embedding the unique identifier into a generated page*" on a per request basis. Claim 16 is directed to a similar method.

Claim 34 is directed to a method for generating a page, including the steps of "*identifying a component class*" of each component and "*storing a first object of the component class, the first object representing the component.*" on a per request basis.

Claim 74 is directed to a software development system using components that are dynamic, since "*the set of components on the generated document can vary for different document requests for the same document template.*"

(IV) Independent Claim 114

Claim 114 is directed to a system for displaying dynamically generated documents. The system includes *a plurality of components for execution on the server, at least one of the components including first features to cooperate with an editor in editing said component.*"

B. Comparison Summary

The examiner has rejected the claims based primarily on a prior patent from Massena. Applicant acknowledges that Massena, like the present invention, allows for the editing of specific components on web page templates. Specifically, Massena uses what he calls a "design time control" ("DTC"), which is a type of component/editor that does not have the interactive functionality of applicant's ISSC's. These two component editors are quite different for many reasons, as will be shown below. Further, other kinds of components contained on web pages were also known before Massena, e.g., java applets or active X controls. However, such components do not provide the interactive editing capability for a functional application that applicant has disclosed and claimed.

There are a number of fundamental differences between DTC's and ISSC's. First, DTC's run as part of the editor during the design of a page and are no longer present at run-time, when the page is requested by the browser of an end-user. ISSC's, on the other hand, are executed at run-time on the server whenever an end-user requests a page. This important distinction is specifically reflected in the claim language used by applicant, as discussed in greater detail below.

For example, the most straightforward example of such claim language is in independent claims 97 and 114, which both include the term "*a plurality of components for execution on the server.*" Independent claim 6 uses a similar variant: "*the server computer further comprising*"..."*components that react interactively on user input by executing instructions on the server.*" Claim 9 is a method for generating documents that recites "*steps for execution on the server.*" **Thus, ISSC's are components which are part of the web application developed, not of the editor (as in DTC's). As components of the application (rather than the editor), ISSC's help to reduce the complexity of dynamic web applications by encapsulating interactive functionality.**

The basic operation of a dynamic web application is to dynamically include/exclude or repeat certain portions of a page. Applicant submits that DTC's can not be contained on such portions because for DTC's, the kind and number of components is fixed during editing. This limitation would seem to restrict the set of application problems which are solvable with DTC's. In contrast, ISSC's do not have this problem. This distinction is specifically reflected in the claim language used by applicant, such as "*set of components on the generated document can vary for different requests*" as recited in claim 5 and 74. For a detailed comparison of ISSC's and DTC's, see section C1 below.

The examiner also alleged that it would be obvious to modify Massena to incorporate teachings from the Pinard patent. However, applicant submits that nothing in such a combination would make DTC's run dynamically as part of a running application. Therefore, the modified DTC's still can not be contained in dynamically included/excluded or repeated portions of a page, as explained in more detail below in section C2.

Therefore, for all the foregoing reasons, applicant submits that ISSC's are fundamentally different than DTC's (including as modified by Pinard) and provide significant advantages for dynamic web applications.

Applicant has also developed a new kind of editor for maintaining the ISSC's on web pages. In contrast, Massena uses a classical editor and extends it by DTC's. Applicant submits that it would not be possible to extend a classical editor for editing dynamic pages containing ISSC's.

Applicant's component-editor is server based rather than client based. This means that once it is installed on a web server, it can be used remotely from every client computer with a suitable browser connected to it. It is well known to the artisan that moving a client based application into an internet setting is a complex and difficult task while opening many new application areas. For additional details see the discussion in section C3 below. This distinction is specifically reflected in the claim language used by applicant, such as "*an editor capable of directly operating on the pages displayed by the browser* " as recited in claim 1, "*an editor operable within the web browser*" as recited in claim 22, "*edit a document displayed by the browser*" as recited in claim 90.

When using applicant's editor, the pages look and ***work identically*** to the end result, except for editing features inserted. In contrast, Massena's editor shows dynamic pages not in working order. In addition, server generated parts of pages are displayed using placeholders in Massena's editor and so they more or less differ in appearance from the end user's view, depending on the amount of dynamic processing done. Thus, for the page developer, editing and debugging can be done inside the inventive editor, while with Massena's editor, the developer of dynamic pages needs to, in violation of the WYSIWYG principle, frequently switch back and forth between the editor and a preview of the application. For more details, see the discussion in section C4 below. This distinction is specifically reflected in the claim language used by applicant, such as "*allowing the user to work on a functional application during development*" as recited in claim 1, and "*editor operates functional applications in an edit mode permitting editing directly in the web browser"* as recited in claim 23.

For all these reasons, applicant submits that the inventive ISSC editor works fundamentally different from and provides significant advantages when compared to a classical editor with integrated DTC's as modified with Pinard.

C. Detailed Comparison

C1. Component DTC and ISSC comparison

Applicant's disclosure is concerned with creating server based web applications that can be used with a web browser. A web browser can send page requests to a web server, which answers the request by sending a page to the browser. In the simplest case, the web server just sends an existing static page stored on the server to the browser. In the case of a server based web application, however, the web server answers a request by asking the web application to dynamically generate a page based on data contained in the request.

Massena's Design Time Controls (DTC) run at the time a web page is designed (called design-time by Massena, see Massena's abstract line 3-5); i.e., as part of an editor (see Massena's figure 1 item 120 inside item 110). When an edited file is saved, run-time text is generated for each DTC directly into the page file (Massena column 5 line 1-10). The resulting file containing the generated run-time text is then saved to a web server. The uploaded pages can then be viewed by end-users without the help of DTC's - it is just the run-time text that is being used. Massena specifically builds in protections to make a browser ignore or to filter out DTC's when a user request is made.

ISSC's, on the other hand, are executed on the server. The editor just places specific tags on the pages to mark components. When a page is requested, the ISSC processor reads a page, finds out about the components marked on it and executes the appropriate components in order to generate browser code. Because this process happens only when a page is requested, all the data contained in a page request can be used to dynamically determine the set of components on a page. It should be obvious that ISSC's need to be present on the server when a page is requested.

Because DTC's generate code during save at design-time, the set of components per page is static for each page request. With ISSC's, on the other hand, the set of components can dynamically vary per page request.

DTC's can be used for static pages since no special processing is needed on the server. ISSC's, on the other hand, are specialized for dynamic pages because of the ISSC-processor and the ISSC's that are executed on the server per page request. DTC's can also be used for dynamic pages by generating run-time text that contains server scripting. However, there is no teaching or suggestion in Massena as to how this is done. The portion of Massena cited by the examiner, i.e., column 5 line 8, does not teach or suggest what this code should look like, or what it should do.

ISSC's are specifically designed to handle interactive communication with the end-user. By way of illustration, we use the term "interactive" to mean that on a first displayed page, an ISSC is shown to the end-user, and the user can interact with it (e.g. by clicking a button component or filling out a field component). The browser then sends information about this interaction as data with a subsequent page request to the server. The present invention makes sure that an ISSC is informed about a user interaction and provides the required data to it. DTC's can be interactive too, but they interact with the developer of a page during editing, as described in Massena at column 3 lines 33-37. In contrast, an ISSC can interact with the end-user. As described by Massena, DTC's use various OLE techniques to communicate with the developer, which is possible because the DTC is working on the same client computer which the developer is using. In contrast, communication with the end-user requires interacting via page generation and page requests. This distinction is specifically reflected in the claim language used by applicant, such as "*components that react interactively on user input by executing instructions on the server*" as recited in claims 3, 6 and 9 and "*component, that can react interactively on subsequent document requests*" as recited in claims 24 and 78.

Server based web applications can be quite complex. ISSC's help to reduce the complexity by structuring the application in a way that encapsulates interactive functionality into components. DTC's, on the other hand, do not occur in the application itself, only the generated code does, so they are not a means for structuring an application.

Dynamic web pages generally use the following operations: insertion of dynamic data into a page (e.g. a counter); dynamically including and excluding parts (e.g. excluding an empty shopping basket); and dynamically repeating parts (e.g. for tabular data received from a database). DTC's cannot be contained on a part of a page that is dynamically included/excluded or that is repeated. Since the generation of run-time text happens when a page is saved in the editor, data from a specific request is not known during generation. Therefore, the generation algorithm cannot decide dynamically based on the data contained in a page request to include, exclude or repeat a component. ISSC's, on the other hand, can handle dynamic inclusion/exclusion and repetition of ISSC's. For interactive components, however, it is not sufficient to just manipulate the visible part of a component. ISSC's make sure that a user can not interact with an excluded component by manipulating the web browser and it makes sure that in the event that a component is repeated, the action of the correction instance is executed.

As a summary, applicant submits that the internal working of ISSC's and DTC's is very different, and that such differences are well represented in applicant's claim language. Further, ISSC's are another way of implementing components on web pages, which improves upon the prior art methods such as DTC's, Active X controls and java applets. Each of these component technologies have their specific advantages and disadvantages.

C2. Modifying Massena's DTC's with Pinard

For a number of claims, the examiner asserted that the combination of Massena and Pinard would render these claims obvious Applicant respectfully disagrees.

Pinard discloses a web based company phone directory. Pinard does not appear to disclose an editor or a software development system (see Pinard Abstract). In the abstract, Pinard mentions a directory application component. This seems to be a specific component of the application which is not contained on a web page. In contrast, DTC's and ISSC's are mechanisms to place a plurality of components on internet pages. According to the examiner's text with respect to claim 34, he cites figure 1 of Massena as disclosing components. Applicant therefore concludes that the examiner interprets DTC's as components, but does not view the directory application component of Pinard in the same way.

In the discussion of the various claims (as discussed in more detail below), the examiner proposes to add features or steps from Pinard to Massena. However, combining selected features with Massena's DTC's can not make them execute at run time. DTC's are inherently made to run at design time, not run time. DTC's interact with the developer using OLE, as disclosed in Massena at column 3, lines 33-37. However, it is impossible for DTC's to interact with the developer at run-time. Furthermore, DTC's interact using a variety of interfaces with a classical editor for web pages, e.g., Frontpage, as disclosed by Massena in column 3, lines 19-21. However, such an editor is not present at run-time. Because DTC's make heavy use of all these client features, there is no way to make the DTC's work at run-time on the server.

Massena's page generation algorithm makes heavy use of the DTC's. However, because DTC's are not available at run-time, simply adding new steps to the page generation algorithm does not make it executable at run-time. In particular, it is not possible to simply add steps that, based on the data of a page request, would exclude components or display them repeatedly, simply because the data of the page request is available at run-time only.

C3. A browser based editor

Web applications have the advantage that they can be used from every computer with a suitable browser connected to a server, without any explicit preparations on the client. Thus, the installation of a web application on a server automatically gives a large number of people access to the application, e.g., installation on a corporate server allows anybody in the company connected access to the application. For example, in the case of the phone directory of Pinard, installation on a server is sufficient to give everybody in the company access. On the other hand, creation of web applications would appear to be much more difficult than the creation of client based programs, because all user interaction must be done using page generation and analyses of data in page requests.

The editor described in applicant's claims is a server based web application. This means that installation of the editor on a corporate server immediately gives everybody access to maintain components on a corporate intranet or internet site. This can be used for communication inside the company, as in groupware applications, or outside the company for content management applications.

In contrast, Massena requires an editor program running in a client based environment and so it must be installed on every client computer as disclosed in Massena at column 3 lines 19-21.

The internal workings of the inventive editor are thus totally different than the classical editor on which Massena's DTC's are based. The inventive editor communicates with the developer by sending generated pages to the developer's browser and by analyzing the data sent back. In contrast, Massena's editor communicates with the developer in the same manner as any other GUI program. This distinction is specifically reflected in the claim language used by applicant, such as "*transforming a first document retrieved from the document store into a* *second document having features which permit editing of the first document*" as recited in claim 26, "*the second software program transmitting, while processing selected requests, second documents to the first software program that make the first software program display a user interface to perform editing functions*" as recited in claim 51 as well as the complete language of claim 125 that inherently interleaves execution and editing.

C4. Comparison of the ISSC editor and DTC

The inventive editor shows pages being edited as looking and working identically to the pages seen by an end user of the developed application except for additional editing features, which are specifically inserted to enable editing. Massena's editor does not appear to show dynamic pages in working order and they do not seem to look **identical** to the pages shown to an end-user. Instead dynamically generated parts are replaced by placeholders, as discussed below.

DTC's use two very different methods for display: one at run-time and the other at design time. At run-time, DTC's are displayed using the generated run-time text. To the extent the run-time text contains server scripts, these are evaluated on the server on a page request as disclosed in Massena at column 5, lines 7-13. DTC's are normal controls as disclosed in Massena at column 3, lines 17-20 and the abstract. As with any other control, DTC's have their native way of displaying themselves at design time.

It is therefore left to the skill and the effort put into the development of a particular DTC by the artisan to make both methods display a more or less similar result. However, at design time, a DTC does not yet know about the results from the server scripts. In fact, a server script may access a database, communicate to other components or access various other resources that are available on the server only. In this event, the DTC at design time can't possibly be displayed in a manner that looks identical to the display at run-time.

There is another principle reason why Massena's editor can not display dynamic pages identical to the page displayed at run-time: page requests for dynamic web pages return possibly different variants of the same page for each page request. Massena's editor, which is a classical editor extended by DTC's according to Massena at column 3, lines 19-21, shows just a single variant. This makes clear that the single variant shown in the editor can not be identical to all the variants of the page shown to the end user. In fact, the more variants differ from each other, the less similar some of the variants become from the special variant shown in the editor. In the case where portions of a page are dynamically included, excluded, or repeated, pages displayed to the developer by Massena's editor would not even be similar to the pages displayed to the end user, because theses operations can produce significant different variants of a single dynamic page.

The MSDN Library (www.msdn.microsoft.com) provides an online reference for various Microsoft products. For example, a description of Microsoft's HTML editor (concurrently submitted in an information disclosure statement), which uses DTC's, explicitly mentions the problem that the editor can not display pages correctly in the presence of server side scripting, when it says:

> Quick view in the HTML editor enables you to:
>
> - View .htm files in a manner similar to how they will look in your browser.
>   - Note Quick view does not process server script, so it may not accurately reflect how .asp files will look in the browser. For example, if a design-time control is created by run-time server script, it will not appear in Quick view.

See the following on-line manual:
http://msdn.microsoft.com/library/default.asp?url=/library/en-us/csedui/html/csdlgquickviewhtmleditor.asp

The claimed editor avoids this problem by allowing the user to edit different variants of the same dynamic page. This is because applications remain working inside the claimed editor. By operating the application inside the editor, the developer can select the desired variant of the page to edit.

Applicant submits that the web application being edited does not stay working inside Massena's editor. For the application to stay working, especially scripts inside the page, links and form-submissions must work. In a working application, links and form-submissions cause a new page to be generated on the server and to be displayed. As is well known, at least some scripts stop working when a page is displayed inside a classical editor. Also, clicking on a link in a classical editor usually edits the link instead of issuing a request. However, even if it requested a dynamic page from the server and edited it, all server side scripting in the dynamic page would be destroyed. Massena adds DTC's to such a classical editor according to Massena at column 3, lines 19-21, but does not seem to modify the function of links, scripts, or form submissions, so the arguments above stay valid for Massena.

In addition, figure 1 of Massena does not show any data flow from the web server back into the editor. Pages from the server are transmitted into the browser, which can be used at run-time as well as provide a preview of a page being edited. There is, however, no data flow in Fig. 1 from the browser into the editor. Since a data flow from the server is required to run a web application, applicant concludes that Massena's editor does not display a running web application during editing, and there is nothing contradictory in Massena's disclosure. This distinction is specifically reflected in the claim language used by applicant, such as "*the editor program comprising first instructions for requesting the document generator to process a dynamic web document leading to a generated document, the editor program further comprising third instructions to modify the dynamic web document*" as recited in claim 59.

III. CLAIMS 1-127 ARE PATENTABLE OVER THE CITED REFERENCES

The examiner rejected claims 1-127 as unpatentable in view of U.S. Patent No. 6,035,119 ("Massena") by itself or in combination with U.S. Patent No. 5,940,834 ("Pinard"). However, applicant respectfully traverses the rejection for the reasons discussed below.

Claim 1

Claim 1 is an broadly stated independent claim directed to a "*software development system*" for a typical online data network, wherein the user (client) computer runs a browser program to interpret web pages posted by the server. The system includes "*a page generator*" and "*an editor*." In direct contrast to Massena, the claimed *page generator* generates web pages that include usable editing features that are specifically included and meant for interpretation by the browser program, not ignored or actively removed as taught by Massena (*see* Massena at col. 5:10-18). In addition, the claimed *editor* operates directly on a page which is displayed and functional in client's browser. In contrast, with reference to Fig. 1, the editor of Massena is used to develop a file 15, which is saved and stored as a web page 135 on the web server 130. There is no teaching or suggestion in Massena that the editor 110 would be operable in the web page retrieved by web browser 160, or that the design time controls described by Massena would provide any such editing functionality for stored web pages. In fact, Massena specifically teaches away from providing such editing functionality for stored web pages. Beginning at column 5, line 11, Massena describes how "only the run-time HTML text or script 140 is processed by the web browser 160" and further, how to ignore or filter out the DTC's, which are commented out of the stored web page so that they cannot be active in the web browser. Also, beginning at column 11, line 61, Massena provides instructions for how to convert the web page back into an editable form so that the author (not the user/browser) can modify and then re-store pages on the web server.

In section C4 above, applicant discusses in detail that Massena's editor does not show pages identical to the pages displayed in the browser. In contrast, claim 1 requires the editor to directly operate on the pages displayed in the browser, i.e., not an identical or even a similar copy, but the actual web page. Applicant further discusses in section C4 that Massena does not show a working application inside the editor. In contrast, the claim requires the user to work on a functional application during development.

The examiner cited figure 1 item 110 as being an editor capable of directly operating on the pages displayed by the browser. Applicant however submits that the arrows on figure 1 display the transmission of the page onto the web server and the transmission from the web server to the browser. In case a page contains server side scripts, as disclosed by Massena at column 5 line 8, these scripts are executed on the web server to dynamically transform the page. Because of this transformation, the page shown in the browser and the page shown in the editor are different. In contrast, claim 1 requires that the editor be capable of directly operating on the page displayed by the browser.

The examiner cited Fig. 3 item 310 of Massena as disclosing the generation of editing features for interpretation by the browser program. Box 310 relates to run-time text or code being generated. Thus, applicant assumes that the examiner interprets editing features as being part of the run-time text or code. At column 5, lines 1-10, Massena distinguishes run-time-information and design-time information. Therefore, the naming and description makes clear that run-time-information is not used during editing, but during final display of a page, while design-time information is used for editing. In addition, in column 5, lines 1-4, Massena discloses that the run-time text is generated only when the page is saved to file by the editor, i.e., after editing. Since it is impossible to use information before it is generated, the run-time text or code generated in Massena can not be used for editing. Applicant therefore concludes that run-time information can not possibly contain information used for editing. Therefore, Fig. 3 box 310 does not demonstrate prior art for generating pages with additional editing information for interpretation by the browser program.

Applicant submits that it is also not possible to interpret "*editing features for interpretation by the browser program*" as part of the design-time information disclosed by Massena. In fact, Massena clearly states that generated pages may contain design-time information, but when interpreted by a browser program, that the design-time information is hidden and only run-time information is interpreted. For example, in column 5, lines 4-13, Massena states that design-time information is ignored by the browser and, in fact, the browser ignores everything but run-time information. In contrast, applicant's claim 1 specifically recites "*editing features for interpretation by the browser program*".

Therefore, applicant submits that Massena does not teach or suggest *additional editing features for interpretation by the browser program*, as recited in claim 1.

Claim 2

Claim 2 is dependent from claim 1. For all the reasons discussed regarding claim 1 above, claim 2 is likewise not taught or suggested by the cited art.

Claim 3

Claim 3 is dependent from claim 2, which is dependent from claim 1. For all the reasons discussed regarding claim 1 above, claim 3 is likewise not taught or suggested by the cited art.

In addition, applicant renews its argument that Massena does not teach or suggest a component which "*reacts interactively on user input by executing instructions on the server*" as recited in claim 3. Instead, Massena provides components that execute their functional capability within the editor, not by going back and interacting with the server. (*See* column 5, lines 1-56). Also, applicant refers to the discussion in section C1 above concerning interactive communication with the user.

Claim 4

Claim 4 is dependent from claim 3, which is dependent from claim 2, which is dependent from claim 1. For all the reasons discussed regarding claims 1 and 3 above, claim 4 is likewise not taught or suggested by Massena. Applicant also renews its arguments from the response to the first office action.

Claim 5

Claim 5 is dependent from claim 3, which is dependent from claim 2, which is dependent from claim 1. For all the reasons discussed regarding claims 1 and 3 above, claim 5 is likewise not taught or suggested by Massena.

Applicant would like to ask the examiner to reconsider the discussion of this claim in the response to the first office action in the light of the new information given in section C1 above concerning dynamic inclusion, exclusion and repetition of components..

In addition, applicant would like to reference Massena at column 5, lines 1-4, which specifically states that the generation of run-time HTML, text or script happens in accordance with the design time control when the web page is saved. Because this generation happens at save time and not later when the page is displayed in a browser, the set of components on a page template is static after the page is saved in Massena. Whenever the page is displayed by a browser, it contains the same set of components. In contrast, applicant claims a dynamic set of components that may dynamically change with every browser request.

Claim 6

Claim 6 is an independent claim directed to a software development system having a page generator and an editor. Applicant has amended claim 6 in the amendment to the first office action.

Applicant would like to ask the examiner to reconsider the discussion of this claim in the response to the first office action in the light of the new information given in section C above.

As discussed in sections C1 and C2 above, DTC's are components that are made to run on the client at design time. DTC's are not needed at run-time as discussed in section C1, and

DTC's use a lot of interfaces available on the client only as discussed in section C2, so it is not possible to run them on a server. In contrast, applicant's claim requires the components to be present on the server.

In addition, section C1 discusses in detail interactive components and why DTC's do not seem to disclose them.

The examiner cited Massena at column 3, lines 33-37, as disclosing components that react interactively on user input by executing instructions on the server. However, applicants reading of the cited portion does not reveal support for the examiner's position. Instead, it appears to applicant that DTC's use design time capabilities provided by OLE to capture user input thereby extending the editor. So obviously, DTC's interact with the developer, but they do not reside in the data store of the server computer nor execute instructions on the server. The cited portion of Massena does not reveal that instructions are executed on the server or that there is some communication going on with the server computer that could make it execute any instructions. In contrast, the claim requires the components to react interactively on user input by executing instructions on the server computer.

Claim 7

Claim 7 is dependent from claim 6. For all the reasons discussed regarding claim 6 above, applicant submits that claim 7 is not taught or suggested by Massena.

Claim 8

Claim 8is dependent from claim 6. For all the reasons discussed regarding claim 6 above, applicant submits that claim 8 is likewise not taught or suggested by Massena.

In addition, the examiner asserts that Massena discloses one component contained within another component. However, as previously noted with regard to claim 4 in the response to the first office action, Massena fails to teach or suggest such a feature.

Claim 9

Applicant has amended the claim, not to distinguish from the cited prior art, but instead to make consistent use of the term "document" and "page".

Claim 9 is an independent claim directed to a method for generating documents using components that react interactively on user input. Claim 9 explicitly states that the steps of the methods are for execution on the server upon a document request. The examiner rejected the claim as being anticipated by Massena and cited column 7, lines 60-67 and column 8, lines 1-10 of Massena in support of his position. However, the cited portion appears to describe steps for execution on the client at design time. All steps described in the cited portion deal with DTC's, especially with their identification. Since DTC's run on the client as detailed in section C1 and DTC's are needed for these steps, it does not seem possible to run the steps on a server. Therefore, Massena does not seem to disclose execution of these steps on the server. Since DTC's run at design-time as detailed in section C1 and document requests occur at run-time, Massena likewise seems not to disclose steps for execution upon a document request. In contrast, applicant's claim requires the steps to be executed on the server, upon a document request. Therefore, applicant submits that Massena does not teach or suggest assigning unique identifiers to components upon a document request.

Applicant believes that assigning unique identifiers to components *upon each document request* in contrast to assigning unique identifiers *statically at design*-time is a great innovation, because this enables a dynamic set of components per document request, a major feature of the invention as explained in section B and detailed in C1.

Claim 10

Claim 10 is dependent from claim 9. For all the reasons discussed with reference to claim 9, as well as to claim 10 in the response to the first office action applicant submits that claim 10 is not taught or suggested by Massena.

The examiner cites column 11, lines 58-60 of Massena in support of his position. Applicant discussed this portion of Massena in the response to first office action. In addition, as detailed in section C1 above, DTC's exist at design-time only, so there are no data objects representing them on the server at run-time. In contrast, applicant's claim requires storing data objects on the server representing at least one of the components.

Claim 11

Claim 11 is dependent from claim 10, which is dependent from claim 9. For all the reasons discussed with reference to claims 9 and 10, applicant submits that claim 11 is patentably distinct from the cited reference.

The examiner cited Massena column 7, lines 60-67 and column 8, lines 1-10 in support of his assertion. Applicant discussed this portion in the response to first office action. Applicant requests that the examiner reconsider applicant's arguments in the light of the new reasoning in section C1 above.

Claim 12

Applicant has amended the claim, not to distinguish from the cited prior art, but instead to make consistent use of the term "document" and "page".

Claim 12 is dependent from claim 11, which is dependent from claim 10, which is in turn dependent from claim 9. For all the reasons discussed with reference to claims 9-11, applicant submits that claim 12 is patentably distinct from the cited reference.

Claim 13

Claim 13 is dependent from claim 11, which is dependent from claim 10, which is in turn dependent from claim 9. For all the reasons discussed with reference to claims 9-11, applicant submits that claim 13 is patentably distinct over the cited reference. In addition, applicant submits that DTC's do not exist on the server at run-time as reasoned in section C1, so it is impossible to call them.

Claim 14

Claim 14 is dependent from claim 11, which is dependent from claim 10, which is in turn dependent from claim 9. For all the reasons discussed with reference to claims 9-11 as well as to claim 14 in the response to the first office action, applicant submits that claim 14 is patentably distinct over the cited reference. In addition, applicant submits that DTC's do not exist on the server at run-time as reasoned in section C1, so it is impossible to call them.

Claim 15

Claim 15 is dependent from claim 11, which is dependent from claim 10, which is in turn dependent from claim 9. For all the reasons discussed with reference to claims 9-11, applicant submits that claim 15 is patentably distinct from the cited reference.

Claims 16-21

Claims 16-21 claim a method for objects stored on the server to communicate with the client computer using document generation and document requests. The examiner rejected these claims in the first office action as being anticipated by Massena. In applicant's reading of Massena, however, DTC's are working on the client computer as explained in section C1 and therefore have no need to communicate with the client computer but can directly interact with the developer working on the client computer. The examiner cited various portions of Massena, however, none of them disclosed analyzing of document requests and triggering actions based on them. This is exactly what is claimed in claim 16. Applicant therefore submits that Massena did not anticipate the claimed method. Applicant would like to ask the examiner to reconsider applicant's response to the first office action in light of discussion of section C1 above and especially the paragraph about interactive components in C1.

Applicant has amended claim 20, not to distinguish from the cited prior art, but instead to make consistent use of the term "document" and "page".

Claim 22

Claim 22 is an independent claim directed to a computer having an editor and a document generator. Applicant has amended the claim, not to distinguish from the cited prior art, but instead to make consistent use of the term "document" and "page".

The examiner asserts that Fig. 1 item 110 of Massena teaches an editor which is operable within the web browser. However, applicant respectfully disagrees. As disclosed by Massena at column 3 lines 19-21, Massena uses a classical editor, which is a program separate from the web browser. Also, Figure 1 illustrates the editor and the browser as two separate boxes. In contrast, applicant's claim recites that the editor is operable within the web browser.

In a prior response, applicant submitted that Massena's invention did not generate documents from templates, but instead generates text for insertion into a document template being edited. In the second office action, the examiner cited Massena figure 3, element 310. Fig. 3 appears to disclose the generation of run-time text; however, figure 3 does not seem to specify what happens with the generated text. On the other hand, in column 5 line 1, Massena specifies that the generated text is inserted into the file being edited. In contrast, applicant's claim recites generating an entirely new document, from a document template.

It is applicant's understanding that insertion of run-time text does not include deletion or modification of other components or text on a document. The latter operations are all possible using applicants invention.

Claim 23

Claim 23 is dependent from claim 22, and for all the same reasons, applicant submits that claim 23 is patentably distinct from the cited reference. In section C4 above, applicant discusses in detail the fact that applications are not functional while being displayed inside the editor used by Massena. In contrast, claim 23 requires the editor to operate functional applications permitting editing directly in the web browser. Applicant therefore submits that claim 23 is patentably distinct over the cited reference.

Claim 24

Claim 24 is dependent from claim 23, which is in turn dependent from claim 22, and for all the same reasons as discussed for those claims, applicant submits that claim 24 is patentably distinct from the cited reference. In addition, applicant refers to the paragraph about interactive components in section C1 as background information. Applicant has amended the claim, not to distinguish from the cited prior art, but instead to make consistent use of the term "document" and "page".

The examiner cited Massena at column 3 lines 33-37, as disclosing components that react on subsequent document requests. In the cited portion of Massena, DTC's communicate with the developer using the editor but not with the end user of the page being edited. This view follows from the fact that DTC's are part of the editor and run at design time, as explained in section C1.

In addition, DTC's directly react on user input without using a document request for transmitting information. This is possible because DTC's run on the client computer and so no means of network communication is required for the components to interact with the user. In contrast, applicant developed component communication with the user using document requests and document generation. Therefore applicant submits that Massena does not teach components that can react on subsequent document requests containing user responses.

Claim 25

Claim 25 is dependent from claim 24, which is in turn dependent from claims 23 and 22, and for all the same reasons as discussed for those claims, applicant submits that claim 25 is likewise patentably distinct from the cited reference. Applicant has amended the claim, not to distinguish from the cited prior art, but instead to make consistent use of the term "document" and "page".

The examiner also asserts that Massena (column 5, lines 50-55) discloses an editor capable of showing a menu of components available for insertion into the document template. However, the cited portion relates to a property browser and property page frame, not to a menu of components as recited in applicant's claim.

In response to the first office action in the parent case, applicant amended the claim to contain an additional restriction that the document generator works upon a document request using component classes to generate browser code. Massena at column 5, lines 1-5 discloses however that generation takes place upon saving a file being edited, i.e., at design time. Document requests however only occur at run-time. Applicant therefore submits that Massena does not disclose document generation using components upon a document request at claimed..

Claims 26-33

Claims 26-33 claim a method to edit a document on a server by transforming it into a second document, by inserting handles or scripts, so that displaying the second document in an appropriate browsing program allows the user to edit the first document. The examiner rejected these claims as anticipated by Massena, but Massena does not present a specialized editing technology. Instead, Massena uses DTC's in a classical editor as disclosed at column 3, lines 19-

21. As detailed in section C3 above, classical editors work on the client computer and directly communicate with the developer - they do not generate documents with scripts or handles to display them in a browsing program to do the editing. For this reason and all the detailed reasoning given in the response to the first office action, applicant submits that claims are patentable over the cited reference.

Claim 31

Claim 31 is dependent on claim 30, which is in turn dependent from claims 26, and for all the reasons discussed regarding claim 26 above, and with regard to claims 26 and 30 in response to the first office action in the parent case, applicant submits that claim 31 is not taught or suggested by Massena.

The examiner cited column 4 lines 30–35 of Massena as disclosing scripts that are generated specifically for the second document and encapsulate information which is incorporated in the first document. The cited portion discloses that there are scripts, but does not disclose that scripts are used to encapsulate information in the first document. In contrast, claim 31 requires that the scripts are used to encapsulate information of the first document.

Claim 32

Claim 32 is dependent from claim 26, and for all the reasons discussed regarding claim 26 above, applicant submits that claim 32 is not taught or suggested by Massena.

The examiner cited column 4 lines 30–35 of Massena as disclosing incorporation of information regarding the first document into the second document. Applicant's reading of the cited portion does not reveal any discussion about a first or a second document nor about incorporation of information. The cited portion therefore seems to be irrelevant for the claim.

Claim 33

Claim 33 is dependent from claim 32, which is in turn dependent from claim 26, and for all the reasons discussed regarding claim 26 and 32 above, applicant submits that claim 33 is likewise not taught or suggested by Massena.

The examiner cited column 1 lines 52–54 of Massena as disclosing incorporation of information into the second document and sending change requests for documents to the server. The cited portion does not appear to discuss incorporation of information into a document nor sending change requests for a document to the server. The cited portion therefore seems to be irrelevant for the claim.

Claim 34

Claim 34 is an independent claim directed to a method for generating a document from a document template having components. Applicant has amended the claim, not to distinguish from the cited prior art, but instead to clarify that the word "representing" refers to the object rather than the class and to make consistent use of the term "document" and "page".

The examiner asserts that one with ordinary skill in the art at the time the invention was made would have been motivated to modify Massena to incorporate teachings from Pinard and thereby obtain the claimed invention. According to Pinard's abstract, a main static aspect and a dynamic aspect are provided. In the dynamic aspect the web page generator generates the directory web pages dynamically on an as requested basis, as stated in line 9 of the abstract. Applicant submits that Massena can not be integrated with the dynamic aspect of Pinard, because this would require the DTC's to run at run-time, which is not possible for various reasons as given in the third paragraph of section C2.

The examiner proposed to modify Massena's generation algorithm by adding a step of "*storing a first object of the component class representing the component.*" Applicant submits however that the claim requires a step of "*storing a first object of the component class representing the component* *based on the data contained in a request initiated by the browser*". Such a step needs the data contained in the document request. This data is not available at design-time. Massena's generation algorithm, however, needs to run at design time, as reasoned in section C2. Therefore, adding a step as claimed to Massena's algorithm would result in an non-functional algorithm. Applicant therefore submits that Massena could not be modified as proposed by the examiner.

The examiner also asserts that Pinard at column 5, lines 58-62, discloses identifying a component class and means of storing objects of the component class. The cited portion appears to discuss a class of items. Applicant's claims recite classes of components. The cited portion does not seem to mention identification of such a class nor does it mention storing of objects of that class. Applicant submits that Pinard does not disclose identifying a component class and means of storing objects of the component class.

In addition, applicant's claim requires that the objects being stored represent components wherein those components are denoted on the document templates. The examiner did not cite any part of Pinard nor Massena as disclosing this feature.

The examiner suggested that someone would have been motivated to modify Massena with Pinard, because classifying templates and storing them improves reusability. Applicant's claim does not talk about classification. The word "class" in the term component class is used to denote the "class" concept used in object oriented programming languages. Therefore, component class refers to the part of the program part that implements a certain component.

Claim 35

Claim 35 is dependent from claim 44, which in turn is dependent on claim 34, and for all the reasons discussed for claims 34 and 44, applicant submits that claim 35 is not taught or suggested by Massena alone or in combination with Pinard.

In addition, the examiner asserts that Massena (column 5, lines 58-62) discloses that the constructor method is used to generate browser code and that it is called during document generation. However, applicants' reading of the cited portion reveals no support for the examiner's position. Instead, the cited portion appears to disclose how controls are created. In column 5, lines 1-2, Massena states that the browser code is generated during the save operation which certainly after a control was created. In contrast, applicant claims that browser code is generated by the constructor and that objects are actually created during document generation.

Claim 36

Claim 36 is dependent from claim 34, and for all the reasons discussed for claim 34, applicant submits that claim 36 is likewise not taught or suggested by Massena alone or in combination with Pinard.

The examiner cited Pinard at column 6, lines 20-25 as disclosing for all components having a name attribute looking up the component object in session memory based on said name attribute. According to applicants reading, Pinard 6:20-25 describes a data base look up operation of employee data and the insertion of that data into a page template. This makes applicant believe that the examiner identifies employee data base entries with components. Applicant submits that data base entries are well understood to be different than components in the state of the art.

The examiner cited Pinard 6:20-25 discloses looking up the component object in session memory. Applicant's reading of the cited portion reveals no support for the examiner's position. The cited portion seems to disclose looking up data from a data base. Applicant submits that there are significant differences between a data base and session memory as described in applicant's specification. Session memory is newly created for each user viewing documents while database records are globally available for every user. It is certainly possible to simulate session memory using a data base; however, this is not done by Pinard. Employee data base entries are generally available before a user starts viewing documents in contrast to session memory being newly created per user. It is also the purpose of employee data base entries to be shared between users retrieving documents in contrast to objects in session memory that are private per user. Applicant therefore submits that modifying Massena with Pinard would not be functional because component objects of multiple users could be confused. In addition, applicant submits that modifying Massena with Pinard would not result in the claimed invention, because session memory and the fact the component objects are stored therein is not disclosed in the cited art.

Claim 34 requires components to be denoted on the document and dependent claim 36 requires that the name attribute be part of the component. Applicant's claim therefore requires the name attribute to be denoted on the document. The examiner cited Pinard at column 6, lines 20-25, as disclosing, for all components having a name attribute, looking up the component object based on the name attribute. The cited portion seems to disclose looking up data from a database. It does not seem to disclose what search criteria is used for the lookup. However, in line 27, employee names given are merely examples, which makes applicant conclude that the template mentioned by Pinard is independent of the actual employee data records stored in the database. A template of Pinard therefore can not contain an employee name or the content of other data fields. In contrast to that, applicants claim requires the data to be searched for be a component name denoted on the document.

Claim 37

Claim 37 is dependent from claim 34, and for all the reasons discussed for claim 34, applicant submits that claim 37 is likewise not taught or suggested by Massena alone or in combination with Pinard. Applicant has amended the claim, not to distinguish from the cited prior art, but instead to make consistent use of the term "document" and "page".The examiner cited column 6, lines 20-30 of Pinard as disclosing generating a unique identifier and pointed to the file names. However, applicants reading of the cited portion does not support the examiner's position. Instead, Pinard makes explicit that the web page file names are retrieved from the data base and not generated as required by the claim.

The examiner further cited this portion of Pinard as disclosing the assignment of a unique identifier to the object and pointed to the file names. Applicant's reading of the cited portion does not support the examiner's position. In the discussion of claim 36, the examiner suggested that the component object is stored in the data base. The cited portion does not, however, disclose storing any data into the data base, as it would be required to assign the unique identifier to the component object.

Claim 38

Claim 38 is dependent on claim 37, which in turn is dependent on claim 34. For all the reasons discussed regarding to claims 34 and 37, claim 38 is likewise not taught or suggested by Massena alone or in combination with Pinard.

The examiner cites Pinard at column 6, lines 23-25, as disclosing the method specified in claim 38. The cited portion seems to disclose a different method for inserting data from the database into a generated page. The claimed method calls methods of component objects whose unique identifiers are contained in the form data set. The cited portion does not seem to mention the form data set. Instead, the cited portion talks about inserting file names into the page template. Applicant therefore submits that the cited portion does not disclose calling a method for all components whose unique identifier occurs in the form data set.

Claim 39

Claim 39 is dependent on claim 34 . For all the reasons discussed regarding to claim 34, claim 39 is likewise not taught or suggested by Massena alone or in combination with Pinard. Applicant has amended the claim, not to distinguish from the cited prior art, but instead to make consistent use of the term "document" and "page".

The examiner relies on the arguments of the previous rejection solely based on Massena to reject the claim again. The claim however includes a step of identifying the component class associated with the component. In the discussion of claim 34, the examiner states that Massena does not disclose identifying a component class and therefore referred to Pinard. The claim also requires calling the constructor of the component class, which is an implementation of storing a first object of the component class. In the discussion of claim 34 the examiner stated that Massena does not disclose storing a first object of the component class. It seems contradictory that the examiner could reject this claim based solely on Massena while the claim introduces and describes matter that the examiner has agreed are not disclosed by Massena.

Claim 40

Claim 40 is dependent from claim 39, and for all the same reasons, applicant submits that claim 40 is likewise not taught or suggested by Massena.

Claim 41

Claim 41 is dependent from claim 1, and for all the same reasons. applicant submits that claim 41 is likewise not taught or suggested by Massena.

Claim 42

Claim 42 is dependent from claim 41, which in turn is dependent on claim 1, and for all the same reasons, applicant submits that claim 41 is likewise not taught or suggested by Massena.

The examiner cited Massena at column 11, lines 57-60 in support of the rejection. Applicant respectfully notes that the language used by Massena, namely: "Once a server hosts a web page, the web page is ready for downloading to a client" (col. 11, lines 59-60) does not teach or suggest the provision of a client-based editor that during the course of execution on the client automatically downloads instructions from the server, as recited in claim 42. Massena simply discloses downloading of web pages from a server. Some of those pages may have built in functionality as a result of being constructed with DTC's, but they do not have editing capability, nor do they download editor instructions from the server. Applicant's claim 42 requires downloading instructions as part of the editor. Massena alone or in combination with Pinard does not teach or suggest downloading of instructions as part of the editor.

Claim 43

Claim 43 is dependent from claim 26, and for all the same reasons discussed therein, applicant submits that claim 43 is likewise not taught or suggested by Massena.

Claim 44

Claim 44 is dependent on claim 34. For all the reasons discussed above regarding claim 34, claim 44 is likewise not taught or suggested by Massena.

Claim 44 further details the step of storing an object as introduced in claim 34. Applicant has amended the claim, not to distinguish from the prior art but to clarify that the step of "storing the first object" refers back to the step of "storing a first object" introduced in claim 34.

The examiner cited column 2, lines 4-5 of Pinard as disclosing the step of storing an object by creating a new object as necessary. However, the cited portion of Pinard appears to relate to storing page templates, not components of document templates. In contrast, claim 34 states that the object being stored must represent a component and is created based on the data contained in a document request. This is clearly different than storing page templates. Applicant therefore submits that Pinard does not disclose the step of storing an object by creating a new object.

Claim 45

Claim 45 is dependent on claim 34. For all the reasons discussed regarding claim 34, applicant submits that claim 45 is likewise not taught or suggested by Massena. Applicant has amended the claim, not to distinguish from the cited prior art, but instead to make consistent use of the term "document" and "page".

The examiner cited Pinard at column 2, lines 4-5 as disclosing document templates where components are denoted using tag syntax. However, the cited portion seems to reveal that page templates can contain fields for storing specific information, but it does not specify what syntax is used to denote these fields. Therefore, applicant submits that the cited reference does not teach or suggest components that are denoted on document templates using tag syntax.

Claim 46

Claim 46 is dependent on claim 45, which in turn is dependent on claim 34. For all the reasons discussed regarding claims 34 and 45, applicant submits that claim 46 is likewise not taught or suggested by Massena.

The examiner cited Pinard at column 2, lines 18-25, as disclosing document templates having components that are denoted using tag syntax, wherein the tag name identifies a component class. However, the cited portion seems to reveal inserting information from the database into a page template. The cited portion does not seem to specify tag syntax and does not seem to reveal details about how the tag name is used. Applicant therefore submits that the cited reference does not disclose or teach components that are denoted on document templates using tag syntax whereby the tag name identifies a component class.

Claim 47

Claim 47 is dependent on claim 36, which in turn is dependent on claim 34. For all the reasons discussed regarding claims 34 and 36, applicant submits that claim 47 is likewise not taught or suggested by Massena. Applicant has amended the claim, not to distinguish from the cited prior art, but instead to make consistent use of the term "document" and "page".

The examiner cited Pinard at column 2, lines 18-25, as disclosing document templates on that components are denoted using tag syntax wherein the tag name identifies a component class.

However, the cited portion seems to reveal inserting of information from the database into a page template. The portion does not seem to specify tag syntax and does not seem to reveal details about how the tag name is used. Applicant therefore submits that the cited reference does not disclose or teach components that are denoted on document templates using tag syntax whereby the tag name identifies a component class.

Claim 48

Claim 48 is dependent on claim 36, which in turn is dependent on claim 34. For all the reasons discussed regarding to claim 34 and claim 36, applicant submits that claim 48 is likewise not taught or suggested by Massena.

The examiner cites Pinard at column 2, lines 5-15, as disclosing reusing the component object, if found to store the first object. Applicant respectfully submits that the examiner is in error reaching this conclusion. The cited portion seems to reveal retrieving information from a database and inserting it into page templates. In contrast, the claim requires the object that was looked up from session memory according to claim 36 to be reused in a specific way. Applicant submits that the cited portion does not disclose reusing an object in session memory as required by the claim.

Claim 49

Claim 49 is dependent on claim 36, which in turn is dependent on claim 34. For all the reasons discussed regarding claims 34 and 36, applicant submits that claim 49 is likewise not taught or suggested by Massena.

The examiner cites Pinard at column 2, lines 5-15, as disclosing creating a new object and storing it in session memory. Applicant respectfully submits that the examiner is in error reaching this conclusion. The cited portion seems to reveal retrieving information from a database and inserting it into page templates. In contrast, the claim requires creating a new object and storing it in session memory. Applicant submits that Pinard does not disclose creating a new object and to storing it in session memory.

Claim 50

Claim 50 is dependent on claim 49, which in turn is dependent on claim 36 and claim 34. For all the reasons discussed regarding claim 34, claim 36 and claim 49, applicant submits that claim 50 is likewise not taught or suggested by Massena by itself or in combination with Pinard.

The examiner cited Pinard at column 2, lines 1-25, as disclosing that new objects are created for all objects not having a name attribute. The step of creating a new object was introduced in claim 49, and claim 50 specifies in more detail when this step is performed. Applicant submits that for the same reasons as in claim 49, creating a new object in session memory is not disclosed by Pinard.

Claim 51

Claim 51 is an independent claim. Applicant could not find locate any reasoning specifically directed at claim 51 in the last office action; and therefore submits that the Examiner has failed to make a prima facie case that the claim is unpatentable. The reasoning for claim 34 is wholly inapplicable to claim 51.

As discussed in section C2 above, applicant's editor works as a web application, while Massena's editor is client based. The claim expresses this by interpreting the editor as a second and third software program that communicate via document requests with a first software program interpreted as a web browser.

Claim 51 requires a second software program communicating with a first software program using document requests and document transmissions. Applicant asserts that none of the cited portions in Massena or Pinard disclose anything like the second software program. The second software program is transmitting second documents, while processing selected requests, to the first software program that make the first software program display a user interface to perform editing functions for maintaining components on document templates.

Claim 52

Claim 52 is dependent from claim 51, and for all the reasons discussed therein, applicant submits that claim 52 is likewise not taught or suggested by Massena alone or in combination with Pinard.

Claim 53

Claim 53 is dependent from claim 52, which in turn is dependent on claim 51. For all the reasons discussed regarding claims 51 and 52, applicant submits that claim 53 is likewise not taught or suggested by Massena alone or in combination with Pinard.

The Examiner cited Fig. 1 of Massena as disclosing a second software program on the server, the second software program transmitting, while processing selected requests, second documents to the first software program that make the first software program display a user interface to perform editing functions. Applicant submits that Fig. 1 of Massena shows that editing takes place in the editor 110 and not the browser 160. The browser interpreted as the first software program therefore does not display an interface to perform editing functions and consequently the web server interpreted as the second software program does not send documents to the browser that make it display a user interface for editing. Instead, Fig. 1 seems to disclose an editor running completely on the client computer. Massena at column 3, lines 19-21, supports this view by stating that Massena extends a classical client side editor. Applicant therefore submits that either Massena alone or in combination with Pinard does not disclose an editor running as a second program on the server computer.

Claim 54

Claim 54 is dependent from claim 52, which in turn is dependent on claim 51, and for all the reasons discussed regarding claims 52 and 51, applicant submits that claim 54 is likewise not taught or suggested by Massena alone or in combination with Pinard.

The Examiner cited Fig. 7 as showing that second documents are HTML pages with embedded scripts. Applicant respectfully submits that the examiner is in error in reaching this conclusion. As reasoned above for claim 53, applicant submits that Massena does not disclose second documents sent to the browser to make it display a user interface for editing.

Claim 55

Claim 55 is dependent from claim 52, which in turn is dependent on claim 51 and for all the reasons discussed regarding claim 52 and claim 51, applicant submits that claim 55 is likewise not taught or suggested by the cited art.

The examiner cited Massena's Fig. 3 as disclosing editing functions including adding, removing, or modifying attributes of a component. Applicant respectfully disagrees. Applicant could not find any of these operations in Fig. 3 and therefore submits that Fig. 3 is irrelevant for this claim.

Claim 56

Claim 56 is dependent from claim 52, which in turn is dependent on claim 51 and for all the reasons discussed regarding claim 52 and claim 51, applicant submits that claim 56 is likewise not taught or suggested by any of the cited art.

Applicant assumes that the examiner interprets the first software program as a web browser and the second software program as a web server. The examiner cited Massena figure 3 as disclosing a fifth software program used by the web server including instructions for generating generated documents from document templates. Applicant respectfully disagrees. In applicant's view of figure 3, box 310 seems to disclose instructions that are part of a DTC which in turn is part of the editor as reasoned in section C1 above. In contrast, the claim requires the fifth program to be used by the web server, which is clearly not the case for the editor. Applicant therefore submits that Massena does not disclose or teach claim 56.

Claim 57

Claim 57 is dependent from claim 56, which in turn is dependent on claim 52 and 51. For all the reasons discussed regarding claim 51, 52 and 56, applicant submits that claim 57 is likewise not taught or suggested by Massena nor by Pinard.

Applicant submits that for all the reasons given with respect to claim 1 Massena does not disclose editing features for interpretation by the first software program.

Claim 58

Claim 58 is dependent from claim 56, which in turn is dependent on claim 52 and claim 51. For all the reasons discussed regarding claim 51, 52 and 56, applicant submits that claim 58 is likewise not taught or suggested by Massena nor by Pinard. Applicant has amended the claim, not to distinguish from the cited prior art, but instead to make consistent use of the term "document" and "page".

The examiner cited figure 1, editor 110 as disclosing instructions that allow the user to click on the generated document to thereby select items upon which to perform edit functions on. Applicant submits, however, that the editor 110 can not possibly show the generated page to the user since Massena's generation happens only when the file shown in the editor is saved. Massena states this in column 5 line 1. In addition, further generation may go on inside the web server since the run-time text might contain server side scripting as stated in column 5 line 8. Based on the arrows shown in figure 1, applicant submits that the editor 110 does not have access to the generated page and therefore can not display it to the user. Consequently it can not allow the user to click on it as claimed.

Claim 59

Claim 59 is an independent claim. The examiner referred to claim 34 for reasoning. However, applicant submits that claim 59 is concerned with editing documents while claim 34 is not, and the examiner's reasoning given for claim 34 is not concerned with editing. therefore, the reasoning for claim 34 is not applicable for claim 59.

As discussed in section C4 above, the claimed editor displays a functional application during editing. Claim 59 expresses this by requiring the editor to comprise instructions for requesting that the document generator process a dynamic document as well as instructions to modify the dynamic document. In contrast Massena's editor does not have this functionality.

Claim 59 requires that the editor program include instructions for requesting that the document generator process a dynamic web document leading to a generated document. According to Massena at column 5, lines 1-10, the page generator of Massena is built into the editor, so there is no request sent from the editor to the page generator. In addition, the generation step in Massena is done when a file is saved. This makes clear that the code generated is not displayed during editing for modification as claimed, but the generated code is used for run-time only.

In addition, the claim requires that the editor work with the dynamic web document as well as with the generated document at the same time, because items contained on the generated document are to be displayed and because the dynamic web document is to be modified. The examiner did not cite anything in Massena that discloses this. Instead, column 5, lines 1-10, make clear that Massena works on a single document only and that generation in Massena means to insert run-time text into this document. The run-time text is not displayed by the editor but just at run-time in the browser.

Claim 60

Claim 60 is dependent from claim 59, and for all the same reasons, applicant submits that claim 60 is likewise not taught or suggested by Massena alone or in combination with Pinard.

In addition, the examiner cites a Massena figure, without giving a number. Applicant assumes that the examiner refers to figure 1 as in claim 61. As explained in section C1 above, page generation is running on the client in Massena. In contrast, claim 60 explicitly states that document generation must run on the server computer.

Pinard's dynamic aspect discloses a page generator running on the server; however, applicant submits that Pinard can not be combined with Massena as reasoned with respect to claim 34 and in section C2. In addition, figure 1 of Massena makes clear that there is no information flow from the web server back into the editor. Therefore, applicant submits that Massena modified with Pinard do not disclose a document generator that is requested by the editor as claimed in claim 59 and that is executed on the server as claimed in claim 60.

Claim 61

Claim 61 is dependent from claim 60, which in turn is dependent on claim 59 and for all the reasons discussed regarding claim 59 and claim 60, applicant submits that claim 61 is likewise not taught or suggested by Massena nor by Pinard.

The examiner cites Massena figure 1 item 110 as disclosing that the document generator contains instructions to collect edit information. Please refer to applicant's reasoning concerning claim 1 for the discussion of edit information.

Claim 62

Claim 62 is dependent from claim 60, which in turn is dependent on claim 59 and for all the reasons discussed regarding claim 59 and claim 60, applicant submits that claim 62 is likewise not taught or suggested by Massena nor by Pinard.

The examiner cites Massena figure 1 item 110 as disclosing an editor that displays said view. However, applicant reasons that according to claim 59, the view needs to display items requested from the server. In applicants view of figure 1, it clearly shows that the requested information is displayed in the browser and not in the editor 110. Applicant therefore submits that neither Massena nor Pinard disclose displaying information of the generated and requested document inside the editor.

Claim 63

Claim 63 is dependent from claim 60, which in turn is dependent on claim 59 and for all the reasons discussed regarding claim 59 and claim 60, applicant submits that claim 63 is likewise not taught or suggested by Massena nor by Pinard.

The examiner cites Massena figure 1 item 110 as disclosing a software development system that can automatically repeat requesting the document generator to process the dynamic web document if required. In applicants view, Massena figure 1 shows that page requests are initiated by the browser and not by the editor. In addition, there is nothing in figure 1 that suggests that requests are somehow issued automatically. In contrast, applicant's claim specifically recites that the editor automatically issues requests.

Claim 64

Claim 64 is dependent from claim 59 and for all the reasons discussed regarding claim 59 applicant submits that claim 64 is likewise not taught or suggested by Massena nor by Pinard.

In applicants view, step 510 of figure 5 of Massena does not seem to disclose generation of browser code. Instead, it reads design time information. Applicant therefore submits that the cited portion appears to be irrelevant for claim 64.

Claim 65

Claim 65 is dependent from claim 64, which in turn is dependent on claim 59 and for all the reasons discussed regarding claim 59 and claim 64, applicant submits that claim 65 is likewise not taught or suggested by Massena nor by Pinard.

According to claim 59, the view must be capable of containing items that can be selected for modification functions. Pages displayed inside a browser do not seem to contain items the user can select in order to apply modification functions. Instead, according to column 3, lines 19-21, Massena uses a classical editor like Frontpage that has its own method of displaying the page being edited. Applicant therefore submits that in Massena the editor does not use a web browser for displaying said view.

Claim 66

Claim 66 is dependent from claim 64, which in turn is dependent on claim 59 and for all the reasons discussed regarding claim 59 and claim 64, applicant submits that claim 66 is likewise not taught or suggested by Massena nor by Pinard.

Claim 67

Claim 67 is dependent from claim 59, and for all the reasons discussed regarding claim 59, applicant submits that claim 67 is likewise not taught or suggested by Massena nor by Pinard.

Applicant submits that in case of Massena, the document displayed in the editor and the document displayed to the end user in the browser do not in all cases look similar as explained in section C4 above.

Claim 68

Claim 68 is dependent from claim 59, and for all the reasons discussed regarding claim 59, applicant submits that claim 68 is likewise not taught or suggested by Massena nor by Pinard.

Claim 69

Claim 69 is dependent from claim 68, which in turn is dependent on claim 59 and for all the reasons discussed regarding claim 59 and claim 68, applicant submits that claim 69 is likewise not taught or suggested by Massena nor by Pinard.

Claim 70

Claim 70 is dependent from claim 69, which in turn is dependent on claim 68 and claim 59. For all the reasons discussed regarding to claim 59, 68 and 69, applicant submits that claim 70 is likewise not taught or suggested by Massena nor by Pinard.

The examiner did not make any specific citation with respect to this claim, and nothing about position information.

Claim 71

Claim 71 is dependent from claim 59, and for all the reasons discussed regarding claim 59, applicant submits that claim 71 is likewise not taught or suggested by Massena nor by Pinard.

As reasoned in section C4 above and as disclosed in column 3, lines 19-21, Massena uses a classical editor for editing but not a web browser. According to claim 59, the view must contain information items the user can select in order to apply a modification function to it. Therefore, the view must be the view displayed by the editor and not the view displayed by the browser as a preview. In contrast, the claim requires that the user can click onto a document displayed by the browser in order to select items for modification.

Claim 72

Claim 72 is dependent from claim 71, which in turn is dependent on claim 59. For all the reasons discussed regarding claims 59 and 71, applicant submits that claim 72 is likewise not taught or suggested by Massena nor by Pinard.

The examiner cited Pinard at column 2, lines 1-20, as disclosing instructions for initiating a reload in the browser. Applicant respectfully submits that the examiner is in error reaching this conclusion. The cited portion lists various features of Pinard; however, none of these include programming the browser in any way, especially not programming the browser to automatically initiate a reload. Applicant therefore submits that the cited portion is irrelevant for claim 72.

Claim 73

Claim 73 is dependent from claim 59, and for all the reasons discussed regarding claim 59, applicant submits that claim 73 is likewise not taught or suggested by Massena nor by Pinard.

The examiner cited Pinard at column 2, lines 10-15 as disclosing an editor program to provide instructions for displaying specific information. Applicant submits that the examiner is in error reaching this conclusion. The cited portion seems to be concerned with retrieving data from the database and filling the data into a web page template. In contrast, the claim is concerned with displaying certain information inside the editor. Applicant therefore submits that the cited portion does not disclose or teach the features of claim 73.

Claim 74

Claim 74 is an independent claim. According to section C1, DTC's can only handle a static set of components while ISSC's can handle a dynamic set. The claim explicitly requires a dynamic component set. Section C2 above reasons that modifying Massena with Pinard still does not lead to system handling a dynamic component set.

The examiner states that Pinard further discloses a software development system having an editor and components that cooperate with the editor and refers to the reasoning of claim 34 and also to Pinard at column 2, lines 1-25. Applicant submits that the examiner is in error reaching this conclusion. In applicants reading of the cited portion of Pinard, Pinard does not disclose a software development system nor an editor nor a plurality of components on document templates. Instead, Pinard seems to disclose the implementation of a specific application, a phone directory, by generating web pages. In contrast, claim 74 requires a software development system, an editor, and components.

In the rejection of claim 34, the examiner proposed to modify Massena with Pinard, adding certain features from Pinard to Massena. Massena does not disclose a set of components on the generated document that can vary for different document requests for the same document template. This can be concluded from Massena at column 5, lines 1-10, which states that the generation is performed when a page is saved to file since this implies that the set of component is static with every request for this file. Applicant further asserts that Massena could not be extended by someone skilled in the state of the art at the time the invention was made to handle a set of components that varies per document request. In order to handle a varying set of components, document generation needs to be done on a per request basis, which implies it has to be done on the server. Applicant now assumes that the examiner interprets components as being the same as Massena's DTC's. However, DTC's run at design time (according to Massena's abstract and section C1) and are needed for the generation algorithm. This means that the generation algorithm could not be moved to run-time. Performing the generation algorithm at design-time implies a static set of components. In contrast, claim 74 requires a dynamic component set.

Applicant further submits that Massena's editor can not handle a set of components that varies per request. The editor edits a page and then afterwards saves it to the server. However, it is only afterwards that the browser can request a page, and so varying a set of components can be known only then. This makes it impossible for the editor to display a correct set of components during editing. In contrast, the claim requires an editor that can handle a dynamic component set.

Claim 75

Claim 75 is dependent from claim 74, and for all the reasons discussed regarding claim 74, applicant submits that claim 75 is likewise not taught or suggested by Massena nor by Pinard.

The examiner cited Pinard at column 2, lines 1-25 as disclosing that the edit function could be adding, modifying and deleting a component. Applicant submits that the examiner is in error reaching this conclusion. Pinard does not seem to disclose an editor at all. The cited portion seems to disclose an algorithm to generate pages with certain steps. In contrast, applicant's claim states that the user interface of the editor provides certain functionality.

Claim 76

Claim 76 is dependent from claim 74, and for all the reasons discussed regarding claim 74, applicant submits that claim 76 is likewise not taught or suggested by Massena nor by Pinard.

The examiner referred to claim 36 for reasoning supporting the rejection. Claim 36 does not seem to talk about tag syntax, but claims 45, 46 and 47 do. For all the reasons discussed concerning claims 46 and 47, applicant submits that claim is likewise not taught or suggested by Massena nor by Pinard.

Claim 77

Claim 77 is dependent from claim 74, and for all the reasons discussed regarding claim 74, applicant submits that claim 77 is likewise not taught or suggested by Massena nor by Pinard.

The examiner cited Massena figure 1 as disclosing a document generator running on the server. However, applicant submits that in Massena document generation is performed as part of the editor on the client computer, as stated in Massena column 5, lines 1-10.

Claim 78

Claim 78 is dependent from claim 74, and for all the reasons discussed regarding claim 74, applicant submits that claim 78 is likewise not taught or suggested by Massena nor by Pinard.

The examiner cited Pinard column 2 lines 33-36 as disclosing excluding a component from the generated document. Applicant submits that the examiner is in error in reaching this conclusion. Pinard does not seem to disclose components on document templates at all as discussed with regard to claim 74, and the cited portion does not seem to disclose excluding of anything. In addition, the cited portion does not seem to talk about a second or subsequent requests as required by the claim. Applicant therefore submits that claim 78 is not taught or suggested by Pinard.

Claim 79

Claim 79 is dependent from claim 78, which in turn is dependent on claim 74. For all the reasons discussed regarding claim 74 and claim 78, applicant submits that claim 79 is likewise not taught or suggested by Massena nor by Pinard.

The examiner cited Pinard at column 2 lines 33-36 as disclosing preventing excluded components from reacting on subsequent document requests. Applicant submits that the examiner is in error in reaching this conclusion. Pinard does not seem to disclose preventing of any execution. In addition, the cited portion does not seem to talk about a second or subsequent requests as pointed out in the claim. Applicant therefore submits that claim 79 is not taught or suggested by Pinard.

Claim 80

Claim 80 is dependent from claim 79, which in turn is dependent on claim 78 and 74. For all the reasons discussed regarding claim 74, 78 and claim 79, applicant submits that claim 80 is likewise not taught or suggested by Massena nor by Pinard.

The examiner cited Pinard column 2 lines 33-36 as disclosing preventing excluded components from reacting on subsequent document requests. Applicant submits that the examiner is in error in reaching this conclusion. The cited portion seems to disclose creating data gathering and formatting mechanisms, generating a list of members in web format and sending the result to a requesting unit. The cited portion does not talk about subsequent document requests nor about remembering selected information in session memory between two requests. Applicant therefore submits that the cited portion is irrelevant for claim 80.

Claim 81

Claim 81 is dependent from claim 74. For all the reasons discussed regarding claim 74 applicant submits that claim 81 is likewise not taught or suggested by Massena nor by Pinard.

The examiner also referred to the reasoning of claim 78. For all the reasons discussed regarding claim 78. applicant submits that claim 81 is likewise not taught or suggested by Massena nor by Pinard.

In addition, claim 81 requires nesting of components. Applicant submits that Massena does not disclose nested components as discussed with respect to claim 4 and that Pinard does not seem to disclose components on document templates at all as discussed with respect to claim 74.

Claim 82

Claim 82 is dependent from claim 74. For all the reasons discussed regarding claim 74, applicant submits that claim 82 is likewise not taught or suggested by Massena nor by Pinard.

As discussed in the reasoning of claim 74 and in section C4 above, a classical editor as used by Massena (see column 3 lines 19-21) is not capable of taking a set of components into account that varies per document request.

Claim 83

Claim 83 is dependent from claim 74. For all the reasons discussed regarding claim 74, applicant submits that claim 83 is likewise not taught or suggested by Massena nor by Pinard.

The examiner referred to the reasoning of claim 78. For all the reasons discussed regarding claim 78, applicant submits that claim 81 is likewise not taught or suggested by Massena nor by Pinard.

In addition, claim 83 requires inclusion of additional component instances, which is not discussed in claim 78. Applicant submits that Massena does not disclose that the editor is capable of showing multiple instances of a single component.

Claim 84

Claim 84 is dependent from claim 74. For all the reasons discussed regarding claim 74, applicant submits that claim 84 is likewise not taught or suggested by Massena nor by Pinard.

The examiner cited Pinard at column 2 lines 1-15 as disclosing that a generated document can contain more components than the document template. Applicant respectfully submits that Pinard does not disclose anything like components on document templates, as discussed with reference to claim 74. As required by claim 74, components have to be capable of cooperating with the editor, which is not the case with the things proposed by Pinard, since Pinard did not disclose an editor at all. In addition, it does not seem possible to run DTC's per page request, since DTC's are made to run on a client computer as reasoned in section C1.

Claim 85

Claim 85 is dependent from claim 74. For all the reasons discussed regarding claim 74, applicant submits that claim 85 is likewise not taught or suggested by Massena nor by Pinard.

The examiner cited Massena at column 3 lines 44-45 as disclosing that a generated document could include multiple instances of a single component denoted on a document template. Applicant however respectfully submits that the examiner is in error reaching this conclusion. The cited portion discloses that instantiate means creating an instance of an object in memory. In contrast, the word instance is used inside the claim in its natural meaning, namely that for a component denoted on the page template code is generated multiple times. This is quite different from creating an object in memory. Applicant therefore submits that claim 85 is likewise not taught or suggested by Massena nor by Pinard.

Claim 86

Claim 86 is dependent from claim 85 which is in turn dependent on claim 74. For all the reasons discussed regarding claim 74 and claim 85, applicant submits that claim 86 is likewise not taught or suggested by Massena nor by Pinard.

The examiner referred to the reasoning of the claim 37 rejection. For all the reasons discussed regarding claim 37 above, applicant submits that claim 86 is likewise not taught or suggested by Massena nor by Pinard.

In addition, claim 86 requires names generated into the browser code to be qualified with the unique identifier. The examiner cited Pinard at column 6, lines 20-25 with respect to claim 37. This portion discloses inserting specific database information into browser code; it does not seem to disclose qualification of names with a unique identifier. Applicant therefore submits that qualification of names inside browser code is not disclosed by Massena nor by Pinard.

Applicant submits that qualification of names with a unique identifier is required in case of multiple instances in order to make sure names in browser code are unique as required by HTML.

Claim 87

Claim 87 is dependent from claim 74. For all the reasons discussed regarding claim 74, applicant submits that claim 87 is likewise not taught or suggested by Massena nor by Pinard.

Applicant submits that Pinard does not disclose components as used in the claim, and further, that neither Massena nor Pinard disclose nested components as discussed with respect to claim 4.

In addition, the claim requires the fourth component to contain instructions that decide how many instances of the fifth component are to be created. The cited portion does not seem to disclose instructions that control how many instances of a component are created. Applicant therefore submits that neither Massena nor Pinard disclose components controlling the number of instances of another component included into a generated document as required by claim 87.

Claim 88

Claim 88 is dependent from claim 74. For all the reasons discussed regarding claim 74, applicant submits that claim 88 is likewise not taught or suggested by Massena nor by Pinard.

As reasoned in section C4 above, the editor used by Massena can only handle a static set of components per page.

The examiner cited Pinard at column 4 lines 57-67 as disclosing that the editor can provide an editable view that includes multiple instances of components. Applicant submits that the examiner is in error reaching this conclusion. The cited portion does not seem to talk about an editor at all. The cited portion therefore seems to be irrelevant for the claim. Applicant submits that neither Massena nor Pinard disclose an editor that can display multiple instances of a single component denoted on a page template in an editable view.

Claim 89

Claim 89 is dependent from claim 74. For all the reasons discussed regarding claim 74, applicant submits that claim 89 is likewise not taught or suggested by Massena nor by Pinard. Please refer to section C4 above for background information. Applicant has amended the claim, not to avoid the cited reference, but to clarify that "displaying" in this context actually means "generation of browser code for displaying."

According to Massena at column 5 lines 1-10, a DTC is a control generating run-time text. This run-time text is used to display the component during normal use of the component, i.e., when a browser accesses the page. During editing, the control is displayed using the normal mechanism of displaying controls. This implies that in Massena, DTC's use different instructions for displaying the component during editing and during normal use. In contrast, applicant claims that the same instructions can be used. Applicant therefore submits that Massena does not disclose using the same instructions as recited in claim 89.

It is of tremendous benefit to use the same instructions, because in this way, a component author just needs to write one routine and not two routines to display the component. In addition, using the same code for display makes sure that a component looks virtually identical during editing as well as during normal use.

Claim 90

Claim 90 is an independent claim. As reasoned in section C3 above, applicant's editor runs as a web application and displays the documents being edited inside a browser. In contrast, Massena uses a classical editor, as disclosed at column 3 lines 19-21.

The examiner cited Massena figure 1 item 110 as disclosing an editor for use with a browser, the editor comprising a first software program for execution within the browser providing the user interface of the editor. Applicant respectfully submits that the examiner is in error. In applicant's view of figure 1, the editor is clearly drawn outside the browser. Since the user interface of the editor is part of the editor, the interface of the editor is outside the browser in Massena as well. In contrast, applicant's claim requires it to be inside the browser. Applicant has amended the claim to make clear that the first software program is provided for processing the user's clicks thereby initiating editing functions.

In addition figure 1 shows that a page displayed in the editor is saved to a server. Afterwards the browser can request the page from the server. Massena column 5 line 1-2 however states that run-time text is generated into the page during the save operation. Massena column 5 line 8 in addition states that pages may contain server side scripting, which implies that the page is modified by server side scripts when being requested by the browser. This means that the browser and the editor show different pages in Massena. In contrast claim 90 requires the editor to work on the document displayed by the browser.

In addition, figure 1 does not seem to disclose that scripts in the document being edited remain functional.

Claim 91

Claim 91 is dependent from claim 90. For all the reasons discussed regarding claim 90 applicant submits that claim 91 is likewise not taught or suggested by Massena nor by Pinard.

Claim 92

Claim 92 is dependent from claim 90. For all the reasons discussed regarding claim 90 applicant submits that claim 92 is likewise not taught or suggested by Massena nor by Pinard.

Claim 93

Claim 93 is dependent from claim 92 which in turn is dependent on claim 90. For all the reasons discussed regarding claim 90 and 92, applicant submits that claim 93 is likewise not taught or suggested by Massena nor by Pinard. Likewise, for all the reasons discussed regarding claim 1, applicant submits that editing features are not taught nor suggested by Massena nor by Pinard.

Claim 94

Claim 94 is dependent from claim 93 which in turn is dependent on claims 92 and 90. For all the reasons discussed regarding claims 90, 92 and 93, applicant submits that claim 94 is likewise not taught or suggested by Massena nor by Pinard.

Claim 95

Claim 95 is dependent from claim 94 which in turn is dependent on claim 93, 92 and 90. For all the reasons discussed regarding claims 90, 92, 93 and 94, applicant submits that claim 95 is likewise not taught or suggested by Massena nor by Pinard.

The examiner cited Massena figure 1 as disclosing a document generator running on the server computer. In applicants view of figure 1, the editor is running on the client computer. According to Massena at column 5 lines 1-2, page generation is performed as part of the editor in Massena. Applicant therefore concludes that page generation is performed on the client in Massena. In contrast, claim 95 requires the page generator to work on the server computer.

Claim 96

Claim 96 is dependent from claim 90. For all the reasons discussed regarding claim 90, applicant submits that claim 96 is likewise not taught or suggested by Massena nor by Pinard.

The examiner cited Massena at column 7 lines 1-5 and the word "dynamic." However, the cited portion does not contain the word dynamic. The cited portion talks about network connections which seems to be unrelated to the claim.

Claim 97

Claim 97 is an independent claim. The examiner referred to claim 34 for reasoning in rejecting the claim. For all the reasons discussed above regarding claim 34, applicant submits that claim 97 is likewise not taught or suggested by Massena. Please refer to section C1 above as background information.

In the reasoning of claim 34, the examiner states that Massena discloses a method for generating a page for display in a browser from a page template containing components and cites figure 1. Since figure 1 clearly shows a design time control, applicant assumes that examiner interprets DTC's as components. The examiner then reasons that Massena could be modified with Pinard. Applicant submits that in order to modify DTC's to work as claimed in claim 97, they need to run on the server computer and they need third program instructions for execution on the server, which are initiated by the user interacting with a component. As explained in section C1 above, however, DTC's must run on the client computer and would not work on the server.

In addition, neither Pinard nor Massena seem to disclose third program instructions for execution on the server which are initiated by the user interacting with a component, nor fourth program instructions on the server computer for analyzing data contained in a document request.

Claim 98

Claim 98 is dependent from claim 97. For all the reasons discussed regarding claim 97, applicant submits that claim 98 is likewise not taught or suggested by Massena nor by Pinard.

The examiner referred to the reasoning of claim 87. Applicant respectfully submits that there is no relationship between claim 87 and claim 98. Applicant believes that claim 98 is concerned with analyzing document requests as discussed with respect to claim 16 and claim 11 and with interactive components as explained in section C1. In contrast, claim 87 is concerned with nested components.

Claim 99

Claim 99 is dependent from claim 98, which in turn is dependent on claim 97. For all the reasons discussed regarding claim 98 and claim 97, applicant submits that claim 99 is likewise not taught or suggested by Massena nor by Pinard.

The examiner cited Pinard at column 2 lines 1-15 as disclosing at least one second component denoted on at least one of the document templates. Applicant submits that the examiner is in error reaching this conclusion. The cited portion talks about multiple fields on page templates. However, neither Pinard's fields not Massena's DTC's (as explained in section C1 about interactive components) seem to include third program instructions for execution on the server which are initiated by the user interacting with the component as required by claim 97. Therefore, it does not seem possible to interpret either DTC's or fields as components. Applicant therefore submits that Pinard does not teach at least one second component denoted thereon.

Claim 100

Claim 100 is dependent from claim 99, which in turn is dependent on claim 98 and 97. For all the reasons discussed regarding claims 97, 98 and 99, applicant submits that claim 100 is likewise not taught or suggested by Massena nor by Pinard.

The examiner refers to claim 45 for reasoning. Applicant submits that the claim is similar to claims 46 and 47 and submits that for all the reasons discussed regarding claims 45, 46 and 47, claim 100 is likewise not taught or suggested by Massena nor by Pinard.

Claim 101

Claim 101 is dependent from claim 99, which in turn is dependent on claim 98 and 97. For all the reasons discussed regarding claims 97, 98 and 99, applicant submits that claim 101 is likewise not taught or suggested by Massena nor by Pinard.

The examiner refers to claim 47 for reasoning. Applicant submits that the claim is concerned with nested components as discussed with respect to claim 4 and submits that for all the reasons discussed regarding claims 4 and 47, claim 101 is likewise not taught or suggested by Massena nor by Pinard.

Claim 102

Claim 102 is dependent from claim 101, which in turn is dependent on claim 99, 98 and 97. For all the reasons discussed regarding claims 97, 98 and 101, applicant submits that claim 102 is likewise not taught or suggested by Massena nor by Pinard.

As reasoned with respect to claim 99, neither Pinard nor Massena seems to disclose anything on page templates that could be interpreted as a component having third program instructions for execution on the server which are initiated by the user interacting with the component as required by claim 97.

Claim 103

Claim 103 is dependent from claim 98, which in turn is dependent on claim 97. For all the reasons discussed regarding claims 97 and 98, applicant submits that claim 103 is likewise not taught or suggested by Massena nor by Pinard.

As reasoned with respect to claim 99, neither Pinard nor Massena seems to disclose anything on page templates that could be interpreted as a component having third program instructions for execution on the server which are initiated by the user interacting with the component as required by claim 97.

Claim 104

Claim 104 is dependent from claim 103, which in turn is dependent on claims 97 and 98. For all the reasons discussed regarding claims 97, 98 and 103, applicant submits that claim 104 is likewise not taught or suggested by Massena nor by Pinard.

The examiner refers to claim 37 for reasoning. Applicant submits that the claim is concerned with the qualifying of names, which is discussed with regard to claim 86. Applicant submits that for all the reasons discussed regarding to claim 37 and claim 86, claim 104 is likewise not taught or suggested by Massena nor by Pinard.

Claim 105

Claim 105 is dependent from claim 98, which in turn is dependent on claim 97. For all the reasons discussed regarding claim 97 and 98, applicant submits that claim 105 is likewise not taught or suggested by Massena nor by Pinard.

The examiner refers to claim 78 for reasoning. Applicant submits that for all the reasons discussed regarding to claim 78, claim 105 is likewise not taught or suggested by Massena nor by Pinard.

Claim 106

Claim 106 is dependent from claim 105, which in turn is dependent on claim 97 and claim 105. For all the reasons discussed regarding claim 97 and claim 105, applicant submits that claim 106 is likewise not taught or suggested by Massena nor by Pinard. Applicant has amended the claim for clarity by replacing "displayed" with "transferred" since this term is more precise.

Applicant submits that there is nothing like third instructions disclosed in Massena or Pinard as detailed in section C1 about interactive components.

Claim 107

Claim 107 is dependent from claim 98, which in turn is dependent on claim 97. For all the reasons discussed regarding claims 97 and 98, applicant submits that claim 107 is likewise not taught or suggested by Massena nor by Pinard.

Applicant submits that there is nothing like third instructions disclosed in Massena or Pinard as detailed in section C1 about interactive components.

The examiner cited Pinard at column 2 lines 1-15 as disclosing analyzing data. Applicant submits that the examiner is in error reaching this conclusion. The data stems from a document request as required by claim 97. The cited portion seems to disclose filling a page template. In contrast, the claim requires data from a document request to be analyzed. Applicant therefore submits that Pinard does not disclose or teach such a feature.

Claim 108

Claim 108 is dependent from claim 107, which in turn is dependent on claim 97 and claim 98. For all the reasons discussed regarding claims 97, 98 and 107, applicant submits that claim 108 is likewise not taught or suggested by Massena nor by Pinard.

The examiner proposed to add "checking for errors" as disclosed by Leshem to Massena modified with Pinard. Applicant respectfully submits that this combination misses the point. The claim requires that subsequent calling of third instructions be suppressed in case of errors. This additional functionality does not seem to be disclosed by Massena nor by Pinard. Applicant therefore submits that claim 108 is likewise not taught or suggested by Massena nor by Pinard, even if "checking for errors" is added.

In addition, the examiner reasoned that it would have been obvious to one ordinary skill in the art to modify Massena as modified with Lesham because it would have made the system more efficient. Applicant respectfully submits that the examiner is in error reaching this conclusion. Adding error checking adds more instructions to a system and therefore in general makes a system less efficient.

Claim 109

Claim 109 is dependent from claim 98, which in turn is dependent on claim 97. For all the reasons discussed regarding claims 97 and 98, applicant submits that claim 109 is likewise not taught or suggested by Massena nor by Pinard.

The examiner cited the abstract of Pinard as disclosing storing a data object in session memory representing a component. Applicant submits that the examiner is in error reaching this conclusion. Pinard does not talk about anything similar to a session memory. The abstract talks about a directory application component of the web page generator. It does not talk about creating or storing this component. Applicant therefore submits that the cited portion does not disclose storing an object in session memory.

Claim 110

Claim 110 is dependent from claim 109, which in turn is dependent on claim 97 and claim 98. For all the reasons discussed regarding claims 97, 98 and 109, applicant submits that claim 110 is likewise not taught or suggested by Massena nor by Pinard.

The examiner cited Pinard figure 1 "requesting between user and database" as disclosing encapsulation of the third program instructions in a method of data objects. Applicant respectfully submits that the examiner is in error reaching this conclusion. Applicant can not see a relation between "requesting between user and database" and encapsulation. Applicant therefore submits that the cited portion is irrelevant for claim 110.

Claim 111

Claim 111 is dependent from claim 110, which in turn is dependent on claims 97, 98 and 109. For all the reasons discussed regarding claims 97, 98, 109 and 110, applicant submits that claim 111 is likewise not taught or suggested by Massena nor by Pinard.

The examiner referred to the reasoning used with respect to claim 85. Applicant submits that for all the reasons discussed above regarding claim 85, that claim 111 is likewise not taught or suggested by Massena nor by Pinard.

Claim 112

Claim 112 is dependent from claim 109, which in turn is dependent on claim 97 and claim 98. For all the reasons discussed regarding claims 97, 98 and 109, applicant submits that claim 112 is likewise not taught or suggested by Massena nor by Pinard.

The examiner referred to the reasoning used with respect to claim 37. Applicant submits that for all the reasons discussed above regarding claim 37, that claim 112 is likewise not taught or suggested by Massena nor by Pinard.

Claim 113

Claim 113 is dependent from claim 112, which in turn is dependent on claims 97, 98 and 109. For all the reasons discussed regarding claims 97, 98, 109 and 112, applicant submits that claim 113 is likewise not taught or suggested by Massena nor by Pinard.

The examiner referred to the reasoning with respect to claim 37. Applicant submits that for all the reasons discussed above regarding claim 37, that claim 111 is likewise not taught or suggested by Massena nor by Pinard.

In addition, claim 113 talks about analyzing said data for unique identifiers, as in claim 16. Applicant therefore submits that for all the reasons discussed above regarding claim 16, claim 113 is likewise not taught or suggested by Massena nor by Pinard.

Claim 114

Claim 114 is an independent claim. The examiner referred to claim 34 for reasoning. For all the reasons discussed above regarding claim 34, applicant submits that claim 114 is likewise not taught or suggested by Massena.

In the reasoning concerning claim 34, the examiner stated that Massena modified with Pinard is prior art for the claim. However, as reasoned with respect to claim 34 above, and as reasoned in section C2 above, applicant submits that modifying Massena with Pinard does not make DTC's operational on a server.

In addition, claim 114 requires components to include features to cooperate with an editor. Thus, it would not be possible to interpret the directory application disclosed by Pinard as a component, because Pinard does not introduce an editor at all.

Claim 115

Claim 115 is dependent from claim 114. For all the reasons discussed regarding claim 114, applicant submits that claim 115 is likewise not taught or suggested by Massena nor by Pinard.

The examiner cited Massena figure 1 item 110 as disclosing components for execution on the server, at least one of these components including fourth program instructions for passing information to the editor. Applicant submits that figure 1 item 110 shows an editor with a DTC. As discussed in section C1, DTC's do not execute on the server. In contrast, claim 114 requires components to execute on the server. Applicant therefore submits that Massena and Pinard do not teach or suggest claim 115.

Claim 116

Claim 116 is dependent from claim 115 which in turn is dependent on claim 114. For all the reasons discussed regarding claims 114 and 115, applicant submits that claim 116 is likewise not taught or suggested by Massena nor by Pinard.

The examiner referred to claim 115 for reasoning. Applicant submits that figure 1 of Massena does not show information flow from the server to the editor running on the client.

Claim 117

Claim 117 is dependent from claim 115 which in turn is dependent on claim 114. For all the reasons discussed regarding claims 114 and 115, applicant submits that claim 117 is likewise not taught or suggested by Massena nor by Pinard.

The examiner cited Massena figure 1 as disclosing transmission of said information from the server to the client and especially refers to requesting between server and client. Applicant respectfully submits that the examiner is in error reaching this conclusion. Applicants view of figure 1 clearly shows that the result of a request is passed to the browser. In contrast applicants claim 115 requires said information to be passed to the editor. Therefore, applicant submits that Massena does not teach or suggest transmitting said information to the client and passing it to the editor.

Claim 118

Claim 118 is dependent from claim 115, which in turn is dependent on claim 114. For all the reasons discussed regarding claims 114 and 115, applicant submits that claim 118 is likewise not taught or suggested by Massena nor by Pinard.

Claim 119

Claim 119 is dependent from claim 114. For all the reasons discussed regarding claim 114, applicant submits that claim 119 is likewise not taught or suggested by Massena nor by Pinard.

Claim 120

Claim 120 is dependent from claim 115 which in turn is dependent on claim 114. For all the reasons discussed regarding claims 114 and 115, applicant submits that claim 120 is likewise not taught or suggested by Massena nor by Pinard.

Claim 121

Claim 121 is dependent from claim 114. For all the reasons discussed regarding claim 114, applicant submits that claim 121 is likewise not taught or suggested by Massena nor by Pinard.

Claim 122

Claim 122 is dependent from claim 121 which in turn is dependent on claim 114. For all the reasons discussed regarding claim 114 and 121, applicant submits that claim 122 is likewise not taught or suggested by Massena nor by Pinard.

The examiner referred to claim 121 for reasoning and cited figure 1 of Massena. Figure 1, however, does not seem to disclose a page for editing component attributes. Applicant therefore submits that a page for editing component attributes is not taught or suggested by Massena nor by Pinard.

Claim 123

Claim 123 is dependent from claim 114. For all the reasons discussed regarding claim 114, applicant submits that claim 123 is likewise not taught or suggested by Massena nor by Pinard.

The examiner referred to claim 36 for the reasoning supporting his rejection. However, claim 36 does not talk about tag syntax, but claims 45, 46 and 47 do. For all the reasons discussed concerning claim 46 and 47, applicant submits that claim 123 is likewise not taught or suggested by Massena nor by Pinard.

Claim 124

Claim 124 is dependent from claim 114. For all the reasons discussed regarding claim 114, applicant submits that claim 124 is likewise not taught or suggested by Massena nor by Pinard. Further, applicant amended the claim, not for avoiding the cited reference, but to clarify that "displaying" in this context actually means "generation of browser code for displaying."

This claim is also very similar to claim 89. Applicant therefore submits that for all the reasons discussed with regard to claim 89 above, claim 124 is not taught or suggested by Massena nor by Pinard.

Claim 125

Claim 125 is an independent claim. The examiner referred to claim 124 for reasoning in rejecting the claim. Applicant submits that the examiner is in error in referring to this claim. Claim 124 is dependent from claim 114, which is concerned with a system for displaying dynamically generated documents in a data network. Claim 125 describes a method for editing an application that is built using components.

Applicant submits that the claimed method drastically differs from the working of a classical editor as proposed by Massena, as discussed above in sections C3 and C4. As disclosed by Massena in column 5, lines 1-2, page generation takes place when a page is saved to file, i.e., after editing. In contrast, the claimed method has document generation take place at the beginning. In addition, Massena's editor displays a page template, while the claimed editing method displays the generated document. In addition, the claimed method has a step of identifying the selected component in the source code based on a view showing a document generated by the application which does not seem to be disclosed by Massena.

Claim 126

Claim 126 is dependent from claim 125. For all the reasons discussed regarding claim 125, applicant submits that claim 126 is likewise not taught or suggested by any combination of Massena and Pinard.

The claim requires a running step, which includes the document generation according to claim 125, to be repeated during editing. According to Massena at column 5, lines 1-2, page generation happens at the end of editing when the page is saved to file.

Claim 127

Claim 127 is dependent from claim 126 which in turn is dependent on claim 125. For all the reasons discussed regarding to claims 125 and 126, applicant submits that claim 127 is likewise not taught or suggested by any combination of Massena and Pinard.

The examiner cites Pinard at column 2, lines 1-25, as disclosing the collecting of edit information for use by the identifying step. Applicant submits, however, that the examiner is in error reaching this conclusion. Pinard seems to disclose a specific web application for phone directories, but does not seem to disclose any steps to collect information for an editor. In fact, Pinard does not seem to disclose an editor at all. Applicant therefore submits that Pinard does not disclose edit information for use by the identifying step at all.

IV. CONCLUSION

For all the foregoing reasons, applicant submits that the claims are in condition for allowance, and the examiner's favorable reconsideration to that end is solicited. If additional questions remain, the examiner is encouraged to telephone the undersigned.

Respectfully submitted,
DERGOSITS & NOAH LLP

Dated: 11/19, 2002

By:
Richard A. Nebb
Reg. No. 33,540

Four Embarcadero Center, Suite 1150
San Francisco, California 94111
(415) 705-6377 tel
(415) 705-6383 fax
rnebb@dergnoah.com